数理解析研究所講究録
1038 巻 1998 年 135-138

近似根の応用について（その３）

北本卓也[1]

Abstract. 工学においては、連立多変数多項式の全ての根が必要な場合がしばしばある。辞書式順序のグレブナー基底を計算すれば、理論的にはこの問題は１変数多項式の求根に帰着させることができ、DKA 法等を用いて解くことができる。しかしながら、実際には辞書式順序のグレブナー基底はしばしば非常に重い計算となり、実質上計算不可能となる場合も少なくない。そこで本稿では、数値的算法と代数的算法を組み合わせて、与えられた連立多項式の全ての数値根とその原始多項式を求める１つのアプローチについて述べる。

1. はじめに

著者は最近、非線形制御の研究をしている知人より、次の連立多変数多項式系の全ての根が計算できないかどうか相談を受けた。(ただし、c_{ij} は有理数)

$$f_1 = c_{11}x_r + c_{12}x_i + c_{13}x_r x_i + c_{14}y_r y_i + c_{15}{x_r}^2 x_i + c_{16}{x_i}^3 + c_{17}x_i {y_r}^2 + c_{18}x_i {y_i}^2,$$
$$f_2 = c_{21}x_r + c_{22}x_i + c_{23}{x_r}^2 + c_{24}{x_i}^2 + c_{25}{y_r}^2 + c_{26}{y_i}^2 + c_{27}{x_r}^3 + c_{28}x_r {x_i}^2 + c_{29}x_r {y_r}^2 + c_{2a}x_r {y_i}^2,$$
$$f_3 = c_{31}y_r + c_{32}y_i + c_{33}x_r y_i + c_{34}x_i y_r + c_{35}{x_r}^2 y_i + c_{36}{x_i}^2 y_i + c_{37}{y_r}^2 y_i + c_{38}{y_i}^3,$$
$$f_4 = c_{41}y_r + c_{42}y_i + c_{43}x_r y_r + c_{44}x_i y_i + c_{45}{x_r}^2 y_r + c_{46}{x_i}^2 y_r + c_{47}{y_r}^3 + c_{48}y_r {y_i}^2$$

上の多項式系の根が、システムの大域的な安定性に関係があり、従来から、こういう多項式系の根を求める必要があったそうである。いままではニュートン法を用いていくつかの実根を求めていたそうであるが、その場合には全ての根が求まるとは限らないので、システムの大域的な安定性が保証できなくて、論文の査読でいつもその点をつかれているとのことだった。そこで、グレブナー基底を用いて全ての根を求めることはできないかと質問された。

上の式から分かるように、辞書式順序のグレブナー基底を計算した場合、最高８１次の式となり、係数も大きくなることが予想されるので、とても計算できそうに無い。実際、PentiumPro 200Mhz, メモリ 128 Mbyte のマシンの *Mathematica* 3.0 で辞書式順序のグレブナー基底の計算をいちおう試みたが、１０時間たっても答えが出ないので計算をあきらめた。しかしながら、homotopy 法や全次数辞書式グレブナー基底などの方法を組み合わせることにより、結局、全ての数値根とその原始多項式を計算することに成功したのでそれについて報告する。

[1] 筑波大学数学系 kita@math.tsukuba.ac.jp

2. 数値的アプローチ

2.1. homotopy 法

homotopy 法は多変数連立多項式の根を求めるための、数値的算法である。この算法は、次のようなステップで $f_i(x,y,...,z)=0$ の根を求める。

1. $f_i(x,y,...,z)=0\ (i=1,...,n)$ の代わりにより単純な問題 $\tilde{f}_i(x,y,...,z)=0\ (i=1,...,n)$ を解く。
2. $tf_i+(1-t)\tilde{f}_i=0$ を t の微分方程式とみなし、$t=0$ から $t=1$ まで数値的に解く。

2.2. homotopy 法での求根

上の homotopy 法には HOMPACK という FORTRAN でかかれたパッケージがあり、フリーで利用できる。著者は近年、この HOMPACK を f2c を用いて C 言語に変換し、筑波大学佐々木研究室で開発中の数式処理システム GAL にリンクして、GAL から使えるようにしている。そこで与えられた問題を GAL で解かせてみると、１分３０秒程度で答えが返ってきた（使ったマシンは CPU PentiumPro 200Mhz、メモリ 128 Mbyte)。答えから以下のことが読み取れた。

1. $(y_r,y_i)=(0,0)$ の根がいくつかある。
2. $(x_r,x_i,y_r,y_i)=(\alpha,\beta,\gamma,\delta)$ が根ならば $(x_r,x_i,y_r,y_i)=(\alpha,\beta,-\gamma,-\delta)$ も根。
3. いくつかの根が発散している。

上の 1., 2. は与えられた多項式を良く見れば、明らかである。3. については、いろいろ調べた所、どうやら根が８１個はなくて行き場を失った根が発散しているらしいということが分かってきた。始めに述べた通り、与えられた多項式の全ての根を求めることが要求されているので、発散していない根だけを答えたのでは十分でない。答えた根が全ての根であることを保証しなければならない。そこでどうしても、根の数を知る必要が出てきた。

3. 代数的アプローチ

先に辞書式順序のグレブナー基底は、実質的には計算不可能であることを述べてが、項順序を変えれば、この限りではない。実際、問題の多項式系の全次数辞書式順序のグレブナー基底は *Mathematica* 3.0 (CPU PentiumPro 200Mhz, メモリ 128 Mbyte) で簡単に計算することができた。そこで良く知られた次の事実

$$\begin{aligned}\text{根の数} &= \dim(R[x,y,...,z]/f_i(x,y,...,z))\\ &= \text{グレブナー基底の head term で簡約化できない単項の数}\end{aligned}$$

を用い、簡単なプログラムを作って、簡約化できない単項の数を計算すると 43 個となった。$(y_r,y_i)=(0,0)$ の根が７個あることと、$(x_r,x_i,y_r,y_i)=(\alpha,\beta,\gamma,\delta)$ が根ならば $(x_r,x_i,y_r,y_i)=(\alpha,\beta,-\gamma,-\delta)$ も根であることに注意すると、$43=7+18\times 2$ より、本質的には１８個の根が求まれば良いことがわかる。そこで homotopy 法の計算結果を整理し、発散した根を取り除き、Complex conjugate で足りない物を補完すると、１８個の根全てが求まった。

4. 原始多項式

以上で当初の目的は達したわけであるが、次に求まった数値根を根とする多項式を求めてみた。これには次の2つの方法を用いて、別々に計算を行った。

4.1. 方法その1

求まった根を $x_1, x_2, ..., x_n$ とすると、まず

$$(x - x_1)(x - x_2) \cdots (x - x_n)$$

を展開し、次にその係数（浮動点小数）を有理数に変換する。

計算には、*Mathematica* 3.0 (CPU PentiumPro 200Mhz, メモリ 128 Mbyte) を用い、小数点以下 5000 桁まで精度を取って計算を行った。計算時間の詳細は不明である（ 10 分以上 10 時間以下)。

4.2. 方法その2

求まった根を $x_1, x_2, ..., x_n$ とすると、

$$\alpha_0 \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} + \alpha_1 \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \cdots + \alpha_r \begin{bmatrix} x_1^r \\ \vdots \\ x_n^r \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

であるので、上式を満たす整数 α_i $(i = 1, ..., r)$ を格子算法を用いて決定する。計算精度は浮動点小数以下 1000 桁をとった。フリーパッケージ NTL の格子算法ルーチン LLL_XD を用いて計算を行った (CPU DEC Alpha 433 Mhz, メモリ 512 Mbyte)。計算時間は 5 分弱（正確には 284 秒）であった。

4.3. 計算結果

方法1の計算結果と方法2の計算結果が一致することを確認した。計算結果は、大体 250 ～ 300 桁程度の係数をもつ多項式となった。

5. まとめ

本稿では、以下のことを述べた。

- 与えられた多項式系の全ての根を、homotopy 法とグレブナー基底を組み合わせた算法で求めた。
- 求まった数値根をもとに、その原始多項式を格子算法等を用いて計算した。

最後の気づきとして、根の中に偶然とは思えない関係があることを見つけている。これから、与えられた多項式に小さな誤差があるのではないかという気がしている。これについては、問題の提供者と後に討議してみるつもりである。

参　考　文　献

[Kit 93] T.Kitamoto "Approximate Eigenvalues, Eigenvectors and Inverse of a Matrix with Polynomial Entries," Jpn. Indus. Appl. Math., Vol. 11, No. 1, 1994.

[TV 93] Thomas Becker and Volker Weispfenning "Groebner Basis,", Springer, 1993.

[GCL 92] K.O.Geddes, S.R.Czapor, G.Labahn "Algorithms for Computer Algebra", Kluwer Academic Pub., 1992.